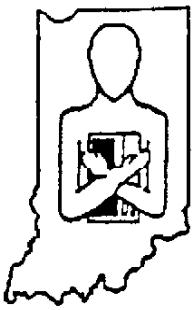

HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP
USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER
GROUP HOOSIER USERS THE HUGgers GROUP HOOSIER USERS
HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP
USERS GROUP HOOSIER HOOSIER USERS GROUP USERS GROUP HOOSIER
GROUP HOOSIER USERS People Helping People GROUP HOOSIER USERS
HOOSIER USERS GROUP HOOSIER USERS GROUP
USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER
GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS

February 1988 THE HUGgers NEWSLETTER Volume 6, Number 1

including December, 1987 and January, 1988

NOTICE

The more time that passes, the more obvious it is to me, and I'm sure to many other members, that I'm no good at being newsletter editor. Therefore, this is my last gasp as editor. I am interested in helping in some capacity with the newsletter, but not as editor. Ironically, while loosing an editor is usually seen as a catastrophic event for a user group, HUG is, in my opinion, getting stronger every month. Right now there are several members who have helped with getting the newsletter out the door. Hopefully, by the time of the next meeting, the problem may be solved (but then again...). I think that part of the solution is the action I have just taken. I know that I've let the group down by missing two month's newsletters. This is a poor way to find out what you're no good at. Oh well, my Apologies to the group and support to my successor.

NOVEMBER MEETING

The November meeting was eventful in that several mambers volunteered as HUG officers for the remainder of the term of office. The names of the new officers can be found on the cover.

In other action, the membership voted to have the status of the treasury printed in the newsletter. The members also voted its intention that we should move toward requiring two signatures on checks written by the user group. The members voted to change the return address of the newsletter to include mention of the fact we are a TI-99/4A club. (As you can see, this hasn't been done yet.)

Do to the length of the business meeting, there was no program.

DECEMBER MEETING

There was no business meeting in December.

Members broke up in several groups for the program features. Jim Ellis had Super Extended Basic in addition to others.

JANUARY MEETING

Members were wondering what had happened to the newletter. Steve Sims gave a report on the treasury.

It was more or less show and tell again for program. I gave a brief demo of Funnelweb v.4 (see Bill Lucid's article). Others were demonstrating other goodies.

THIS MONTH

Presentations are still being lined up for February. Attend the meeting and see what's new.

HUG TREASURY

Balance brought forward:	$288.02
less checking fees:	(2.68)
less Nov. newsletter:	(93.60)
Balance as of Jan. 10, 1988:	$191.74

What follows are the meeting dates that Bob Stahlhut worked out with the people at St. Ann's.

Dec 20, 1987	Apr 17, 1988
Jan 10, 1988	May 15, 1988
Feb 14, 1988	Jun 12, 1988
Mar 13, 1988	

FUNNELWEB 4.0 CONTINUES.....

by Wm. M. Lucid, HOOSIER USER'S GROUP
Indianapolis, Indiana

Documentation is provided for FUNNELWEB version 4.0 in five text files; FWDOC/FASM, FWDOC/LOAD, FWDOC/REPT, FWDOC/TIWR, and FWDOC/UTIL. There are twenty-one pages of documentation, detailing version 4.0. FUNNELWEB version 4.0 is a two disk package from the FAIRWARE authors Tony and Will McGovern; their address, to which contributions for this package may be mailed, is listed in the text file -READ-ME.

FUNNELWEB may be loaded using TI Extended Basic, Myarc XBII, Editor/ Assembler, or TI-Writer. A major change has been incorporated in the LOAD program. LOAD is now a hybrid file of Extended Basic and machine code and must be edited by means of the CONFIG program. CONFIG is documented in the FWDOC/UTIL, but it is largely self-prompting. Use a disk manager to transfer LOAD from disk to disk.

When FUNNELWEB is auto-loaded using LOAD or UTIL1 it searches the disk controller for the drive number used to load FUNNELWEB, and writes this information into system file-names. FUNNELWEB refers to this process as BOOT disk tracking. LOAD/UTIL1 have an escape hatch which disables boot tracking and allows the system drive to be specified directly. CONFIG allows setting of drive numbers to a single character "x" in the form DSKx. for each source disk. Forms such as RD (for Ram Disk) are not supported. The secondary system drive number is for the E/A side of the Central Menu. Screen colors selection is available on almost any screen which invites selection by number, pressing 0 (zero) will cause the screen colors to cycle through the ten color choices. Quick Directory, pressing <FCTN-7>, will call up a disk directory routine with some of the functions used for SD in the Editor. File QD must be on the FUNNELWEB boot disk. Pressing keys <1-9,0> as indicated on the current directory page will cause the Display filename indicated to be marked and written to the mailbox as the workfile name.

Prime focus of FUNNELWEB is the Central Menu and User list screens. The first is TI-Writer, the second programmer's workbench, and third User list options. The user must provide programs for DATABASE and MODEM. The DATABASE dummy filename is DP which would be the filename you need to use for your DATABASE program. If more than one file, the first would be DP, the second DQ, the third DR etc. The same pertains to MD for the MODEM program, first MD, second ME, third MF etc. User has the option of paging the Central Menu and User List screens by pressing the spacebar.

TI-Writer improvements in FUNNELWEB version 4.0. There have been NO change from the original Editor in the maximum size of files that may be loaded or written. Color selection is by using <CTRL-3> or may be altered using CONFIG. Printer device-name is pre-loaded into the Editor by the loader -- it may be changed by CONFIG. A default Edit filename may be set by CONFIG. If left blank, the default DSKx. or the pre-existing filename will be set, where x is the default utility drive number specified by CONFIG. The <FCTN-=> system quit is disabled at all times while in the Editor, including SD. SD handles 127 filenames. Paging is controlled by use of the single <CTRL> or <SHIFT> keys. Pressing <=> causes SD to check and indicate the type of program files on the disk. An asterisk after the filename indicates a fractured file (one stored in more than one continuous block on the disk). Display type files may be marked for the Load File/Save File functions and <V>iew or <D>elF by pressing the number (1-9,0 for 1-10) of the entry as displayed on the current page. Pressing <P> when using SD prints the directory in two column format to the Editor print-device. A right margin warning beep has been incorporated. The inset from the right margin may be set, or the beep canceled , but it requires disk sector editing of the second word in the ED file (after the 3 word file header). It is set to 5 (0005) in distribution versions. A null word here cancels the beep. The EoF marker has been changed to a "position number display", this more like a ruler, as the cursor is not displayed positioned on the line. I find this feature to be of more use than the previous Editor "bottom line".

This package is well documented, performs as documented, and USER customizing of the package has been greatly enhanced by use of the CONFIG program. All users are encouraged to print out the complete documentation to gain optimum use of this package.

(Reprinted from TRI-Valley 99ers' CHANNEL 99 OCT '87)

Modeming on your TI (Part 4 of a 3 part series)
By Greg McGill (Sysop TI-keep BBS [805] 499-5415)

Terminal programs

Last month I went over the various BBS's you will come across in your adventures in modeming, I may have left a few out, due to there aren't very many of that kind of BBS. This month will be an review/recommendation/overview of the major Terminal Programs for your TI.

TERMINAL EMULATOR 2 (TE2)

TI's only FULLY COMPLETED terminal program. This program is SLOW, BULKY and the screen colors are discusting on a TV. Otherwise it is wonderful! The Graphics, Music, Speech, Colors, and word-wrapping are a blessing. This program is set up for only 110 and 300 baud unfortunately. There is a patch floating around that allows you to modify TE2 to do 1200 baud. this patch requires a load/interrupt switch and editor assembler (Funnlweb will work) and a WIDGET is helpful. If you would like this program, just ask at the meeting and I'll give you a copy. I personally only use TE2 for calling TIBBS boards due to the gargantuan amount of music, graphics, speech etc. online there. TE2's file transfer protocol is slow and not real efficient compared to Xmodem. Te2's good point is that it is on cartridge for Easy fast access, otherwise, it's faults outweigh it's good points.

FAST TERM

Paul Charlton's WONDERFUL terminal emulation system. This program is very professional, has many features, including TE2 compatibility as well as Xmodem downloading, this program is also the FASTEST terminal program around. With any other program you get minor delays when it is displaying the incoming data, this SLOWS you down. Fast Term has almost NO delay at all! IT FLYS! I'd suggest printing out the documentation on it and keeping it handy when online, due to some of those Function/shift combo's are a bit tough to remember. I use fast term about 80% of the time. There are also a few MODIFIED versions of fast term floating around, one even has an phone directory/dialer and logon files to auto dial, connect, and logon the board while you watch! My favorite term program, and also the only one for the GENEVE so far! This program will support TE2 Transfers and will ignore all TE2 speech, graphics, and other data while still responding to the BBS's who look for TE2 capability. (Some boards figure out whether your are a TI or not by checking for the TE2 cartridge)

MASS TRANSFER

Mass Transfer is our best PERSON to PERSON term program around. The capability of transferring more than one file at a time is great for trading or sending software over the phone lines to a friend. A MXT bbs has even been designed to have MASS DOWNLOAD capability, but this BBs is quite new and not very wide spread yet. The latest version of MXT (Mass Transfer) has Ymodem file transfer protocol as well as Xmodem. The two protocols are nice due to Ymodem sends very large blocks of data (Good for clear phone lines) and xmodem sends short blocks (Good for a dirty or bad phone line). This program suffers from the same feature TE2 does, Delays slow down your data reception and output, but it's still better than TE2 for more reasons than one. This program will not do any TE2 transfers and will not be recognized as TE2.

4A-TALK

I personally have never used this program. From what I've read, it seems to be almost exactly like Fast term with a few major differences, It comes with a auto-dialer in the original version of 4a-talk, and it is COPY RIGHTED and must be purchased from a dealer or Databiotics. 4a-talk really will do the job for you if you like paying for software. While the fair-ware term programs are just as powerful and almost free.

OMEGA

This program is not yet completed, but, there is a pre-release version available on most bulletin boards. This pre-release version is very interesting, it supports ONLINE RUN LENGTH ENCODED (rle) PICTURE viewing, has a very very nice Xmodem transfer system, user definable macro keys, and an interesting buffering system. I enjoy using this program to call non-ti bbs's where the download section has ASCII downloading, I use the rle mode to display the RLE pictures there, then just hit the Save command on Omega and wala, instant rle download. I am anxiously awaiting the release of the full featured version of this program.

P-TERM

This terminal program is very easy to learn, making it great for beginners, unfortunately it has very few features other than just the usual terminal activities and Xmodem downloads. Not much else to say about Pterm.

If you have anything to add to my opinions and descriptions please feel free to write it all down, and send it to Jeff, the Editor and he will be happy to print your opinions/descriptions in the next newsletter.

My most issued hint/suggestion: READ THE DOCUMENTATION AND EVERYTHING DISPLAYED ON THE SCREEN!

TI WRITER TIPS
by Dennis Sherfy

If you are writing a paper which includes a long word, or complex name, used several times, don't type it over and over. Instead, pick an abbreviation such as xx. Pick a couple of letters which you can type quickly which will never appear together in your letter. Use the abbreviation in place of your long word. Then, when you are finished, use Replace String to substitute the long word for your abbreviation. TI Writer will replace a dozen such words in seconds. The Replace String entry might be: /xx/National Geographic/.

You can use a single letter abbreviation if you set up your Replace String command properly. Use a letter which can never stand alone, such as m. Then write your Replace String command as:

/ m /motion sickness/

This command will replace each m that has a space before and after the letter and will not bother an m used as part of another word. The drawback with the single letter method is that it will not work at the end of a sentence. / m /motion sickness/ will not replace "m.". The computer sees the period as another letter. On the other hand, I can't think of any words containing qq, ww, yy, hh, jj, kk, zz, xx, cc, or vv. You could also use a character like the tilde, but it requires pressing two keys at once, which is probably slower for most people.

SOUTH REGIONAL MEETING

The date for the February South Regional meeting will be announced at the next membership meeting.

DEBUGGING

by Jim Peterson

When you have finished writing a program, the next thing you should do is to run it. And, very probably, it will crash!

Don't be discouraged. It happens to the very best of programmers, very often.

So, the next thing to do is to debug it. And you are lucky that you are using a computer that helps you to debug better than some that cost ten times as much.

There are really three types of bugs. The first type will prevent the program from running at all - it will crash with an error message. The second type will allow the program to run, but will give the wrong results.

And the third type, which is not really a bug but might be mistaken for one, results from trying to run a perfectly good program with the wrong hardware, or with faulty hardware. As for instance, trying to run a Basic program, which uses character sets 15 and 16, in Extended Basic.

First, let's consider the first type. The smart little TI computer makes three separate checks to be sure your program is correct. First, when you key in a program line and hit the Enter key, it looks to see if there is anything it can't understand - such as a misspelled command or an unmatched quotation mark. If so, it will tell you so, most likely by SYNTAX ERROR, and refuse to accept the line.

Next, when you tell it to RUN the program, it first takes a quick look through the entire program, to find any combination of commands that it will not be able to perform. This is when it may crash with an error message telling you, for instance, that you have a NEXT without a matching FOR, or vice versa.

And finally, while it is actually running and comes to something that it just can't do, it will crash and give you an error message - probably because a variable has been given a value that cannot be used, such as a CALL HCHAR(R,C,32) when R happens to equal 0.

The TI has a wide variety of error messages to tell you when you did something wrong, what you did wrong, and where you did it wrong. But, it can be fooled! For instance, try to enter this program line (note the missing quotation mark).
100 PRINT "Program must be saved in merge format."

And, sometimes you may be told that you have a STRING-NUMBER MISMATCH when there is no string involved, because the computer has tried to read a garbled statement as a string.

Also, the line number given in the error message is the line where the computer found it impossible to run the program; that line may actually be correct but the variables at that point may contain bad values due to an error in some previous line.

If the error occurs in a program line which consists of several statements, and you cannot spot the error, you may have to break the line into individual single-statement lines. This is the easiest way to do that - Be sure the line numbers are sequenced far enough apart. Bring the problem line to the screen, put a ! just before the first ::, and enter it. Bring it back to the screen with FCTN 8, retype the line number 1 higher, use FCTN 1 to delete the first statement and the ! and ::, put a ! before the first ::, and continue. Then, when you have solved the bug, just delete the ! from the original line and delete all the temporary lines.

Pages 212-215 of your Extended Basic manual list almost all the error codes, and almost all the causes of each one - it will pay you to consult these pages rather than guessing what is wrong.

You may create some really bad bugs when you try to modify a program that was written by someone else - especially if you add any new variable names or CALLs to the program. Your new variable might be one that is already being used in the program for something else, perhaps in a subscripted array. I have noticed that programmers rarely use @ in a variable name, so I always tack it onto the end of any variable that I add to a program.

Also, the program that you are modifying may have ON ERROR routines, or a prescan, already built in. The ON ERROR routine was intended to take care of a different problem than the one you create, so it could lead you far astray - you had better delete that ON ERROR statement until you are through modifying.

The prescan had better be the subject of another lesson, but if the program has an odd-looking command !@P-up near the front somewhere, it has a prescan built in. And if so, if you add a new variable name or use a CALL that isn't in the program, you will get a SYNTAX ERROR even though there is no error. One way to solve this is to insert a line with !@P+ just before the problem line, and another with !@P- right after it.

When a program runs, even though it crashes or is stopped by FCTN 4 or a BREAK, the values assigned by the program to variables up to that point will remain in memory until you RUN again, or make a change to the program, or clear the

memory with NEW. This can be very useful. For instance, if the program crashes with BAD VALUE IN 680, and you bring line 680 to the screen and find it reads
CALL HCHAR(R,C,CH)
just type PRINT R;C;CH and you will get the values of R, C and CH at the time of the crash. You will find that R is less than 1 or more than 24, or C is less than 1 or more than 32, or CH is out of range.

In Extended Basic, you can even enter and run a multi-statement line in immediate mode (that is, without a line number), if no reference is made to a line number. So, you can dump the current contents of an array to the screen by
FOR J=1 TO 100::PRINT A(J);:: NEXT J - or you can even open a disk file or a printer to dump it to.

You can also test a program by assigning a value to a variable from the immediate mode. If you BREAK a program, enter A=100 and then enter CON, the program will continue from where it stopped but A will have a value of 100.

You can temporarily stop a program at any time with FCTN 4, of course (the manual says SHIFT C, but it was written for the old 99/4), and restart it from that point with CON. Or you can insert a temporary line at any point, such as 971 BREAK if you want a break after line 970. Or, you can put a line at the beginning of the program listing the line numbers before which you want breaks to occur, such as ! BREAK 960,970,980 Note that in this case the program breaks just BEFORE those listed line numbers. You can also use BREAK followed by one or more line numbers as a command in the immediate mode.

The problem with using BREAK and CON is that BREAK upsets your screen display format, resets redefined characters and colors to the default, and deletes sprites. So, it is sometimes better to trace the assignment of values to your variables by adding a temporary line to DISPLAY AT their values on some unused part of the screen. If you want to trace them through several statements, it will be better to GOSUB to a DISPLAY AT. And if you need to slow up the resulting display, just add a CALL KEY routine to the subroutine.

Sometimes, your program will appear to be not flowing through the sequence of lines you intended (perhaps because it dropped out of an IF statement to the next line!) and you will want to trace the line number flow. This can be done with TRACE, either as a command from the immediate mode or as a program statement, which will cause each line number to print to the screen as it is executed. If used as a command, it will trace everything from the beginning of the program, so it is usually better to insert a temporary line with TRACE at the point where you really want to start. Once you have implemented TRACE, the only way to get rid of it is with UNTRACE.

TRACE has its limitations because it can't tell you what is going on within a multi-statement line, and it will certainly mess up any screen display. Sometimes it is better to insert temporary program lines to display line numbers. I use CALL TRACE( ) with the line number between the parentheses, and a subprogram after everything else
30000 SUB TRACE(X)::DISPLAY AT(24,1):X :: SUBEND

Some programmers use ON ERROR combined with CALL ERR as a debugging tool, but I can't tell you much about that because I have never used it. ON ERROR can give more trouble than help if not used very carefully, and I cannot see that CALL ERR gives any information not available by other means.

Sometimes you can debug a line by simply retyping it. It is only very rarely that the computer is actually interpreting a line differently than it appears on the screen, but retyping may result in correcting a typo error that you just could not see. In fact, most bugs turn out to be very simple errors.

When you are debugging a string-handling routine, don't take it for granted that a string is really as it appears on the screen - it may have invisible characters at one or both ends. Try PRINT LEN(M$) to see if it contains more characters than are showing; or
PRINT "*"&M$&"*" to see if any blanks appear between the asterisks and the string.

There is no standard way to debug a program. Each problem presents a challenge to figure out what is going wrong, to devise a test to find out what is really happening.

Don't debug by experimenting, by changing variable values just to see what will happen, etc. Even if you succeed, you will not have learned what was wrong so you will not have learned anything - and if your program contains lines that you didn't understand when you wrote them, you will have real problems if you ever try to modify the program. (Believe me, I speak from experience!)

TI-99/4A COMPUTER OWNERS

If you own a TI-99/4A computer and would like to put it to work for you, the Hoosier User's Group can help you.

The group is made up of TI-99/4A computer owners, some with basic knowledge, some with advanced knowledge. All of them are willing to help others make better use of their computers. The group has a library of over 1,500 programs available to members, has monthly meetings, and publishes a monthly newsletter.

You can obtain help in programming, word processing, using spread sheets, etc.

You can get more information about the Hoosier User's Group by writing

Hoosier User's Group
P.O. Box 2222
Indianapolis IN 46206-2222

or call 881-5918.

TI-99/4A COMPUTER OWNERS

If you own a TI-99/4A computer and would like to put it to work for you, the Hoosier User's Group can help you.

The group is made up of TI-99/4A computer owners, some with basic knowledge, some with advanced knowledge. All of them are willing to help others make better use of their computers. The group has a library of over 1,500 programs available to members, has monthly meetings, and publishes a monthly newsletter.

You can obtain help in programming, word processing, using spread sheets, etc.

You can get more information about the Hoosier User's Group by writing

Hoosier User's Group
P.O. Box 2222
Indianapolis IN 46206-2222

or call 881-5918.

TI-99/4A CLUB
P.O. Box 2222
Indy, IN 46206
881-5918

TI-99/4A CLUB
P.O. Box 2222
Indy, IN 46206
881-5918

TI-99/4A CLUB
P.O. Box 2222
Indy, IN 46206
881-5918

TI-99/4A CLUB
P.O. Box 2222
Indy, IN 46206
881-5918

TI-99/4A CLUB
P.O. Box 2222
Indy, IN 46206
881-5918

TI-99/4A CLUB
P.O. Box 2222
Indy, IN 46206
881-5918

TI-99/4A CLUB
P.O. Box 2222
Indy, IN 46206
881-5918

TI-99/4A CLUB
P.O. Box 2222
Indy, IN 46206
881-5918

TI-99/4A CLUB
P.O. Box 2222
Indy, IN 46206
881-5918

TI-99/4A CLUB
P.O. Box 2222
Indy, IN 46206
881-5918

FIRST CLASS MAIL

TIME DATED
JAN 23 1988
MATERIAL

Don Eicher
9410 Cardinal Drive
Indianapolis, IN 46237

HOOSIER USERS GROUP
P.O. Box 2222
Indianapolis, IN 46206-2222
Forwarding and Address
Correction Requested

HOOSIER USERS GROUP DIRECTORY

OFFICERS

President...Phil Caldwell 1-249-2386
Vice-Pres...Carl Clark 1-398-6276
Secretary...Bob Mikels 925-3821
Treasurer...Steve Sims 631-7255
Regional Center:
South.....Dennis Sherfy 881-5918
Liaison w/ St. Ann's Church & School:
Bob Stahlhut 856-4962

MONTHLY MEETING LOCATION

ST. Ann's School
2839 S. McClure
Indianapolis, IN

Meetings open at 2:00 PM

HUGbbs INFORMATION

317-631-994A 300 baud only
The HUGbbs is on-line 24 hours a day.
SYSOP: Steve Sims - 631-7255 (voice)

NEWSLETTER EXCHANGE

The HOOSIER USERS GROUP is participating in a Newsletter Exchange program with other 11 Users Groups. This offer is made with the understanding that, with proper credit, your Users Group can reprint articles from the HOOSIER USERS GROUP Newsletter, and with proper credit, we can reprint articles from other TI Users Groups Newsletters.

PRINTOUTS

Library listings can be ordered for $.25 6 X 9 self addressed envelope with $.66 postage. The HUGbbs Reference Guide can be ordered for $.50 and a 4 X 9 self addressed envelope with $.22 postage. Please send orders to our P.O. Box. SORRY, PRINTOUTS WILL BE SENT TO ACTIVE MEMBERS ONLY.

BACK ISSUES
ADVERTIZING POLICIES

Write to the P.O. Box number or call one of the officers for details.